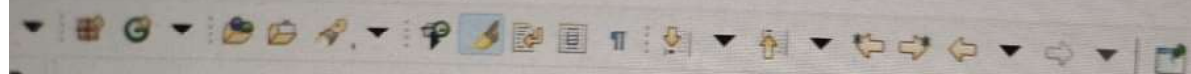


service/impl/PlayListServiceImpl.java - Spring Tools for Eclipse

Project Run Window Help



*EmployeeController.java *EmployeeValidator.java PlayListController.java

```
1 package com.music.playlist.service.impl;
2
3 import com.music.playlist.dto.PlayListRequest;
4
5 @Service
6 public class PlayListServiceImpl implements PlayListService {
7     private PlayListRepository repo;
8
9     public PlayListServiceImpl(PlayListRepository repo) {
10         this.repo = repo;
11     }
12
13     @Override
14     public PlayList createPlayList(PlayListRequest pt) {
15         PlayList p = new PlayList();
16         p.setName(pt.name());
17         p.setTracksCount(pt.tracksCount());
18         return repo.save(p);
19     }
20
21     @Override
22     public List<PlayList> getPlayLists() {
23         return repo.findAll();
24     }
25
26     @Override
27     public PlayList getPlayListById(Long id) {
28         return repo.findById(id).orElse(null);
29     }
30
31     @Override
32     public void deletePlayList(Long id) {
33         repo.deleteById(id);
34     }
35 }
```

oller/PlayListController.java - Spring Tools for Eclipse

ct Run Window Help



*EmployeeController.java *EmployeeValidator.java PlayListController.java x PlayListServiceImpl.java

```
1 package com.music.playlist.controller;
2
3 import com.music.playlist.dto.PlayListRequest;
4
5
6
7
8
9
10
11
12
13
14 @RestController
15 @RequestMapping("v1/playlists")
16 public class PlayListController {
17     @Autowired
18     private PlayListService playListService;
19
20     public PlayListController(PlayListService playListService) {
21         this.playListService = playListService;
22     }
23
24     @PostMapping()
25     public ResponseEntity<PlayList> createPlayList(@RequestBody PlayListRequest playListRequest){
26         return new ResponseEntity<>(playListService.createPlayList(playListRequest),HttpStatus.CREATED);
27     }
28
29     @GetMapping("/{playlistId}")
30     public ResponseEntity<PlayList> getPlayListById(@PathVariable Long playlistId){
31         if(playlistId==null) {
32             return new ResponseEntity<>(HttpStatus.NOT_FOUND);
33         }
34         return new ResponseEntity<>(playListService.getPlayListById(playlistId),HttpStatus.OK);
35     }
36
37
38     @GetMapping
39     public ResponseEntity<List<PlayList>> getAllPlayLists(){
40         return new ResponseEntity<>(playListService.getPlayLists(),HttpStatus.OK);
41     }
42
43     @DeleteMapping("/{playlistId}")
44     public ResponseEntity<Void> deletePlayList(@PathVariable Long playlistId){
45         playListService.deletePlayList(playlistId);
46         return new ResponseEntity<>(HttpStatus.NO_CONTENT);
47     }
48 }
49
50
```

I



Search

Writable

Smart Insert

1:1:0

